Cálculo de Programas

2.º Ano de LCC+LEI (Universidade do Minho)
Ano Lectivo de 2012/13

Teste — 17 de Junho de 2013
09h00
Salas CP2 201, 202, 203 e 204

Importante — *Ler antes de iniciar a prova:*

- *Este teste consta de **10** questões que valem, cada uma, 2 valores. O tempo médio estimado para resolução de cada questão é de 15 min.*
- *Os alunos que entregaram o **miniteste** só podem responder às questões 7, 8, 9 e 10, devendo entregar o teste ao fim de uma hora.*
- *Os restantes alunos devem responder a todas as questões, entregando o teste ao fim de duas horas e meia.*

PROVA SEM CONSULTA (60m / 2h30m)

Questão 1 O combinador

$$\begin{array}{l} \mathsf{const} :: a \to b \to a \\ \mathsf{const}\ a\ b = a \end{array}$$

está disponível em Haskell para construir funções constantes, sendo habitual abreviarmos const k em $\underline{k}$, qualquer que seja k. Demonstre a propriedade

$$\underline{(b,a)} \quad = \quad \langle \underline{b}, \underline{a} \rangle \tag{1}$$

a partir da propriedade universal do produto e de propriedades de funções constantes que conhece, por exemplo:

$$\underline{k} \cdot g \quad = \quad \underline{k} \tag{2}$$

$$f \cdot \underline{k} \quad = \quad \underline{f\ k} \tag{3}$$

Questão 2 É dada uma função α sobre a qual sabe que $\alpha \cdot i_1 = id$ e $\alpha \cdot i_2 = \triangledown$, em que $\triangledown = [id\ , id]$. Infira através de um diagrama a propriedade natural ("grátis") da função α e demonstre-a formalmente.

Questão 3 Sabendo que

$$(p? + p?) \cdot p? \quad = \quad (i_1 + i_2) \cdot p? \tag{4}$$

demonstre a seguinte regra de simplificação de condicionais aninhados:

$$p \to (p \to f, g), (p \to h, k) \quad = \quad p \to f, k \tag{5}$$

Questão 4 Derive, a partir das propriedades que conhece da exponenciação, a definição do operador

$$\mathsf{curry}\ f\ a\ b = f\ (a, b)$$

que consta do Prelude do Haskell.

Questão 5 A função que calcula a lista dos n primeiros números naturais (por ordem inversa),

$$\begin{array}{l} insg\ 0 = [\,] \\ insg\ (n+1) = (n+1) : insg\ n \end{array}$$

pode definir-se em recursividade múltipla com uma função auxiliar:

$$\begin{array}{l} insg\ 0 = [\,] \\ insg\ (n+1) = (fsuc\ n) : insg\ n \\ fsuc\ 0 = 1 \\ fsuc\ (n+1) = fsuc\ n + 1 \end{array}$$

Recorrendo à lei de recursividade múltipla (entre outras) derive desse par de funções a seguinte implementação de $insg$ como um ciclo-for:

$$\begin{array}{l} insg = \pi_2 \cdot insgfor \\ insgfor = \mathsf{for}\ \langle \mathsf{succ} \cdot \pi_1, cons \rangle\ (1, [\,]) \end{array}$$

tal que $fsuc = \pi_1 \cdot insgfor$ e onde $\mathsf{succ} = (1+)$ e $cons\ (n, m) = n : m$. **NB:** Recorde que todo o ciclo-for é um catamorfismo de naturais: $\mathsf{for}\ f\ k = (\![[\underline{k}\ , f]]\!)$.

Questão 6 Mostre que os catamorfismos de naturais $f = (\![[\underline{k}\ , \underline{k}]]\!)$ e $g = (\![[\underline{k}\ , id]]\!)$ são a mesma função.

Questão 7 Pretendendo-se calcular quantos elementos de uma árvore LTree satisfazem um dado predicado p, alguém escreveu

$$\mathsf{length} \cdot (\![[(p \rightarrow singl, nil)\ , conc]]\!) \tag{6}$$

e alguém escreveu

$$(\![[(p \rightarrow \underline{1}, \underline{0})\ , add]]\!) \tag{7}$$

onde $singl\ a = [a]$, $nil = \underline{[\,]}$, $conc = \widehat{(+\!\!+)}$ e $add = \widehat{(+)}$. Mostre, por fusão-cata, que (6) e (7) são a mesma função. (**NB:** assuma as propriedades $\mathsf{length}\ (x +\!\!+ y) = \mathsf{length}\ x + \mathsf{length}\ y$ e $\mathsf{length}\ [a] = 1$.)

Questão 8 Defina-se o anamorfismo

$$suffixes = [\!(g)\!]$$

sobre cujo gene g se sabe o seguinte:

$$g \cdot \mathsf{in} = ! + \langle cons, \pi_2 \rangle \tag{8}$$

Apresente (justificadamente) os passos que faltam na seguinte derivação da versão em Haskell desta função:

$$
\begin{aligned}
& \mathit{suffixes} = [\![g]\!] \\
\equiv \quad & \{ \text{ universal-ana } \} \\
& \mathsf{out} \cdot \mathit{suffixes} = (id + id \times \mathit{suffixes}) \cdot g \\
\equiv \quad & \{ \text{ ... após vários passos ... } \} \\
& \left\{ \begin{array}{l} \mathit{suffixes}\ [\,] = [\,] \\ \mathit{suffixes}\ (h : t) = (h : t) : \mathit{suffixes}\ t \end{array} \right.
\end{aligned}
$$

Questão 9 A propriedade aritmética

$$k \times n \quad = \quad \underbrace{k + \cdots + k}_{n \text{ vezes}}$$

é parafraseada pela igualdade

$$(k\times) \cdot count = sum \cdot (\mathsf{LTree}\ \underline{k}) \tag{9}$$

válida sobre árvores de tipo LTree, onde $count = (\![[\underline{1}\ , add]]\!)$ e $sum = (\![[id\ , add]]\!)$ e $add\ (n, m) = n + m$. Demonstre (9) recorrendo a leis básicas da aritmética e às leis de catamorfismos que conhece (eg. fusão, absorção, etc) instanciadas para LTree.

Questão 10 O functor

$$\mathsf{T}\ X = X \times X$$

oferece um mónade que nos permite trabalhar com pares encarados como vectores a duas dimensões. Por exemplo, neste mónade a expressão

$$\textbf{do}\ \{ x \leftarrow (2, 3); y \leftarrow (4, 5); return\ (x + y) \}$$

dá $(6, 8)$ como resultado — a soma dos vectores $(2, 3)$ e $(4, 5)$.

Fazendo $\mu = \pi_1 \times \pi_2$ e $u = \langle id, id \rangle$, demonstre as seguintes propriedades essenciais à evidência de que o functor dado equipado com μ e u é, de facto, um mónade:

$$\mu \cdot u = id \tag{10}$$
$$\mu \cdot u = \mu \cdot \mathsf{T}\ u \tag{11}$$
$$\mu \cdot \mu = \mu \cdot \mathsf{T}\ \mu \tag{12}$$
